Congratulations! You passed!

Grade received 100%
Latest Submission Grade 100%
To pass 80% or higher

Go to next item

1. A window function is able to access more than the current row of the query result **1 / 1 point**

- ○ No
- ○ Not sure
- ◉ Yes

✓ **Correct**
Correct! A window function is able to access more than the current row of the query result.

2. Which of the following is not a window function? ***(Select all that apply)*** **1 / 1 point**

- ☐ DENSE_RANK()
- ☑ ROLLUP

✓ **Correct**
Correct! This is not a window function.

- ☑ RANK_NUMBER()

✓ **Correct**
Correct! This is not a window function.

- ☐ RANK()
- ☐ ROW_NUMBER()

3. _____ assigns a rank to all the rows and skips the next rank value for two rows with the same rank value **1 / 1 point**

- ○ DENSE_RANK()
- ○ ROW_NUMBER()
- ○ PERCENT_RANK()
- ◉ RANK()

✓ **Correct**
Correct! The RANK() function assigns a rank to all the rows within every partition. Rank is assigned such that rank 1 is given to the first row and rows having the same value are assigned the same rank. For the next rank after two same rank values, one rank value will be skipped.

4. What would be the result of this query? **1 / 1 point**

```
1   SELECT first_name, email, department, salary,
2   NTILE(4) OVER(ORDER BY salary DESC) group_1,
3   NTILE(10) OVER(ORDER BY salary DESC) group_2,
4   NTILE(100) OVER(ORDER BY salary DESC) group_3
5   FROM employees;
```

- ◉ This query retrieves the employees first names, emails, departments, salaries, and three additional columns. Group_1 divides the result set into 4 groups, group_2 divides the result set into 10 groups, and group_3 divides the result set into 100 groups.
- ○ This query retrieves the employees first names, emails, departments, salaries, and three additional records. Group_1 divides the result set into 4 groups, group_2 divides the result set into 10 groups, and group_3 divides the result set into 100 groups.
- ○ This query retrieves the employees first names, emails, departments, salaries, and three additional rows. Group_1 divides the result set into 4 groups, group_2 divides the result set into 10 groups, and group_3 divides the result set into 100 groups.

✓ **Correct**
Correct! This query retrieves the employees first names, emails, departments, salaries, and three additional columns. Group_1 divides the result set into 4 groups, group_2 divides the result set into 10 groups, and group_3 divides the result set into 100 groups.

5. How would you return the running total of the salary for each department ordered by the hire_date? **1 / 1 point**

◉
```
1   SELECT first_name, hire_date, department, salary,
2   SUM(salary) OVER(PARTITION BY department
3                    ORDER BY hire_date) AS running_total
4   FROM employees;
```

```
SELECT first_name, hire_date, department, salary,
TOTAL(salary) OVER(ORDER BY hire_date
                   PARTITION BY department) AS running_total
FROM employees;
```

```
SELECT first_name, hire_date, department, salary,
SUM(salary) OVER(ORDER BY hire_date
                 PARTITION BY department) AS running_total
FROM employees;
```

```
SELECT first_name, hire_date, department, salary,
TOTAL(salary) OVER(PARTITION BY department
                   ORDER BY hire_date) AS running_total
FROM employees;
```

Correct
Correct! This query returns the running total of the salary for each department ordered by the hire_date

6. ROWS is a logical operator used to define the window frame. **1 / 1 point**

- (•) False
- ( ) Not sure
- ( ) True

Correct
Correct! ROWS is not a logical operator but a positional operator to define the window frame.

7. _____ allows you to break the result set into a specified number of approximately equal groups or buckets. **1 / 1 point**

- (•) NTILE()
- ( ) DENSE_RANK()
- ( ) GROUPING SETS()
- ( ) PAGING()

Correct
Correct! The NTILE() window function allows you to break the result set into a specified number of approximately equal groups or buckets.

8. Given a running average taken over this boundary: **"ROWS BETWEEN 2 PRECEDING AND CURRENT ROW".** How many rows will be used to calculate the running average? **1 / 1 point**

- ( ) 4 rows
- ( ) No row
- (•) 3 rows
- ( ) 2 rows

Correct
Correct! The running average will be calculated using the current row and two rows before the current row. This makes it **3 rows** used in the calculation.

9. _____ keyword represents the final row of the partition and is often used as the upper boundary. **1 / 1 point**

- (•) UNBOUNDED FOLLOWING
- ( ) UNBOUNDED PRECEDING
- ( ) UNBOUNDED BELOW
- ( ) UNBOUNDED ABOVE

Correct
Correct! UNBOUNDED FOLLOWING represents the final row of the partition and is often used as the upper boundary.

10. _____ operator produces subtotals and grand totals for every permutation of the columns provided. **1 / 1 point**

- (•) CUBE()
- ( ) ROLLOVER()
- ( ) ROLLUP()
- ( ) GROUPING SETS()

Correct
Correct! The CUBE() operator produces subtotals and grand totals for every permutation of the columns provided.